Chapter 10
Hierarchical MPC-Based Control of an Irrigation Canal

A. Sadowska, P.J. van Overloop, C. Burt, and B. De Schutter

Abstract We discuss the problem of controlling an irrigation canal to accommodate fast changes in the canal state in response to events such as offtakes announced with no time lag or sudden weather changes. Our proposed approach comprises a hierarchical controller consisting of two layers with decentralized PI controllers in the lower layer and a centralized MPC-based event-driven controller in the higher layer. By incorporating the hierarchical controller structure we achieve a better performance than with the PI controllers only as currently in use in the real world, while barely increasing the communication requirements and remaining robust to temporary communication link breakdowns as the lower layer can work independently of the higher layer when the links are being restored. The operation of the higher-layer controller relies on controlling the head gate and modifying the settings of the local controllers. This way, an acceleration of water transporting is attained as the controller allows for rapid reactions to the need for more water or less water at a location. Specifically, when there is a sudden need for water, the storage in some of the pools is used to temporarily borrow water. Alternatively, when there is too much water at a location, it can be stored for some time in upstream or downstream pools before the PI controllers manage to remove the water.

A. Sadowska (✉) • B. De Schutter
Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands
e-mail: a.d.sadowska@tudelft.nl; b.deschutter@tudelft.nl

P.J. van Overloop
Water Resources Management, Delft University of Technology, Delft, The Netherlands
e-mail: P.J.A.T.M.vanOverloop@tudelft.nl

C. Burt
Irrigation Training and Research Center (ITRC), California Polytechnic State University, San Luis Obispo, CA, USA
e-mail: CBurt@calpoly.edu

C. Ocampo-Martinez, R.R. Negenborn (eds.), *Transport of Water versus Transport over Water*, Operations Research/Computer Science Interfaces Series 58,
DOI 10.1007/978-3-319-16133-4_10

10.1 Introduction

10.1.1 Motivation and Contributions

One of the prominent control problems encountered in the area of water management is to control water flows in an irrigation canal swiftly and with little resources involved. This is prompted by the fact that with diminishing world resources of fresh water, it is of a great importance to manage the remaining water supplies in a way that minimizes potential water losses. In terms of control of irrigation canals, it calls for the gates connecting canal pools being operated in an intelligent way, so there are no spills and water is not wasted, but is used for the purpose of crop irrigation as intended. As a matter of fact, fresh water use for agriculture amounts to over 90 % [17] of the overall fresh water use worldwide. Such a large share is undoubtedly partly due to the old equipment that is still widely used in agriculture as any modernizations would need to be most likely paid by the farmers and thus tend to be not well accepted. Consequently, the control schemes that rely on the immoderate utilization of the vulnerable equipment are unfavorable. For instance, as the communication links that are present in the field are often not reliable enough to be used in a continuous control loop, control schemes that depend excessively on communication may fail to be realizable in practice. Nevertheless, despite such limitations, undoubtedly there is a need for efficient, accurate and resource-conscious water management schemes to meet the aforementioned operational criteria.

All this makes some of the existing methods proposed in the literature for controlling an irrigation canal potentially problematic in reality when adverse practical conditions are present. In that respect, the distributed control strategies introduced in, e.g., [1, 5, 9, 15] as well as the centralized control strategies introduced in, e.g., [22, 27, 28] may turn out to rely too heavily on the communication links, thus being rendered impracticable for a real-life application. This is caused by the requirement of these controllers to communicate in every control step during their operation either with each other in case of distributed control, or with a control center in case of centralized control. The other extreme solution broadly used in practice is the application of decentralized PI controllers installed at each gate along the canal [10, 11, 16, 27]. The decentralized PI controllers, while they do not add up to the communication burden as such controllers require only local information from a pool, may be unable to produce a good enough overall performance. In response to such practical restrictions while not compromising on the performance attained, we propose in this chapter a hierarchical control algorithm. The lower control layer is constituted by the existing local PI controllers and as such there is no need for any additional equipment to be installed on top of the one already present in the field. On the other hand, the higher control layer is formed by a centralized controller, designed using principles of Model Predictive Control (MPC) [4, 12] and activated on an event-driven basis, i.e., only when new events occur, e.g., a water delivery request in one of the delivery points along the canal or a heavy

rainfall. In such circumstances, the higher-layer controller is activated and modifies the head gate settings as well as the setpoints of the local PI controllers. Therefore, in normal operating conditions, the PI controllers take full care of the canal. However, even when there is an event, the additional communication required is limited as the higher-layer controller is designed to communicate to each local PI controller only once per activation to convey all required changes. The changes take form of stepwise setpoint changes: once each setpoint is modified from its original value and then it is set back to the normal operating level. This makes the scheme robust to temporary communication losses as in face of the communication links being down temporarily, the PI controllers still autonomously control the canal.

The hierarchical controller discussed in this chapter has been analyzed earlier on in [18] in which differences in system performance when the higher-layer controller is invoked at every control step or in response to events only were studied. Also [19] considers the hierarchical controller, and specifically the controller there is adopted to fit the Central California Irrigation District Main Canal with ramp-shaped setpoint changes ordered by the higher-layer controller as opposed to stepwise setpoint changes discussed here. In that view, the current chapter contributes in a threefold manner. First, we give an improved and unified account of the previous work. Second, we extend the permissible class of the input profiles. Third, we alter the parameters of the lower-layers controllers too, i.e., the control gains of the PI controllers are found through an optimization routine.

10.1.2 Control Problem Description

We now state the particular control problem under consideration. The aim is to control an irrigation canal to allow swift changes, e.g., new deliveries or flow changes in existing offtakes with no time lag between the moments they are announced and start. However, this has to be done ensuring a certain performance level, i.e., not disturbing excessively the normal canal operation. This means, e.g., that the control actions found by the controller should not result in drying out of the canal or water overtopping the canal embankments. If needed, also stricter constraints may be posed with water levels maintained within a tighter range.

To fulfill the control objectives, a two-layer control strategy is introduced. The lower layer consists of the PI controllers, and the higher layer works by altering the head gate settings and by modifying the settings of the local controllers, and is derived in accordance with MPC techniques.[1] The local controllers are influenced

[1]MPC is a model-based control technique that uses predictions of the state and forecasts of the external inputs to determine optimal future control actions for the system. At every step, the control sequence is found through solving an optimization problem in a receding horizon manner. We refer to [4, 12] for a detailed description of MPC.

by the higher-layer controller by their setpoints being altered in a stepwise manner and by the P and I gains being changed. We consider two possibilities for how the changes are handled.

A) The proportional and integral gains are found beforehand as a first stage, and stay fixed thereafter.
B) The algorithm starts with preexisting proportional and integral gains (chosen by the designer). Then, whenever the higher-layer controller is activated, on top of temporary setpoint changes, the gains are also altered pro tem and afterwards return to their original values.

We now briefly discuss what control inputs are required to be found by the higher-layer controller in cases A) and B) above. More details on how the control inputs specifically influence the controllers in the lower layer will be given in Sect. 10.2.2.

In both cases, the head gate settings are altered by the higher-layer controller, i.e., the variable $M_{\text{head gate}} \in \mathbb{R}$ denoting the position of the head gate is changed. This is done with the help of an N_{p}-element sequence

$$\tilde{M}^{\text{control}}_{\text{head gate}} = (M^{\text{control}}_{\text{head gate}}(0), \ldots, M^{\text{control}}_{\text{head gate,}}(N_{\text{p}} - 1))^T, \tag{10.1}$$

where, as per the principles of MPC, N_{p} is the length of the prediction horizon. Also in both cases A) and B) above, the stepwise setpoint modifications of the local PI controllers are assumed and are found using Time Instant Optimization Model Predictive Control (TIO-MPC) [8, 18, 24, 25]. In a nutshell, while in the standard MPC, the N_{p}-element sequence of each control variable is to be found, in TIO-MPC for a prespecified number of control variable changes in a given prediction window of length N_{p}, the optimization routine returns the time instants when the control variable should change and to what value. In that spirit, to characterize stepwise changes of the setpoints, three quantities are needed: the time instant $t_i^{\text{on}} \in \mathbb{R}$ when the setpoint in pool i should diverge from its predefined level $h_i^{\text{ref, normal}}$, the altered value $h_i^{\text{ref, delivery}} \in \mathbb{R}$ of the setpoint, and the time instant $t_i^{\text{off}} \in \mathbb{R}$ when the setpoint should return to the normal operating value $h_i^{\text{ref, normal}}$. For all pools $i = 1, \ldots, N$ we collect the three values in vectors

$$\begin{aligned} H^{\text{ref, delivery}} &= [h_1^{\text{ref, delivery}}, \ldots, h_N^{\text{ref, delivery}}]^T, \\ T^{\text{on}} &= [t_1^{\text{on}}, \ldots, t_N^{\text{on}}]^T, \\ T^{\text{off}} &= [t_1^{\text{off}}, \ldots, t_N^{\text{off}}]^T. \end{aligned} \tag{10.2}$$

Note that by modifying the setpoints using TIO-MPC in the specific way that each setpoint is changed twice, i.e., once to modify the setpoint from its normal levels to deal with an event and the second time to return to the prespecified value, we limit the amount of interference of the higher-layer controller with the local controllers, which results in diminished communication requirements. As explained earlier, in the alternative formulation of the problem using the standard MPC, direct optimization of the setpoints $h_i^{\text{ref}}(j)$ for the whole prediction horizon could be done. However, to impose exactly two changes of the setpoints, integer constraints

would need to be added [2], resulting in an escalation of the computational burden. In contrast, by using TIO-MPC, the problem is not posed as a mixed integer programming problem but as a real-valued optimization problem and thus the computational requirements are also reduced.

In contrast to $\tilde{M}^{\text{control}}_{\text{head gate}}$, $H^{\text{ref, delivery}}$, T^{on} and T^{off}, the proportional and integral gains are changed differently for the two cases. A common part is a preliminary calibration step done by the designer to choose initial values of the gains. In case A) these values are kept later on with no modifications. Conversely, in a steady-state operation in case B), the predefined values are used whereas when the higher-layer controller is activated to deal with an event, the gains are changed in a stepwise manner similarly to the setpoint, and can be different for each event. Therefore, the controller needs to provide modified values of the gains $K_{\text{P},i}$ and $K_{\text{I},i}$, i.e., to find $K^{\text{temp}}_{\text{P},i}$ and $K^{\text{temp}}_{\text{I},i}$, and to determine the time instants $T^{\text{on,gains}}$ when to switch to the modified values and the time instants $T^{\text{off,gains}}$ when to switch back to the predefined values.

10.1.3 Outline

The outline of this chapter in as follows. In Sect. 10.2 we first describe the dynamical model of a canal in general and for the specific case study; next the proposed solution is discussed, i.e., the improved and extended hierarchical control approach is introduced. Further, in Sect. 10.3 we illustrate how the control approach works in a simulation based case study involving the Central California Irrigation District Main Canal. In Sect. 10.4 we discuss how the proposed strategy for transport of water can be combined with a complementary strategy for transport over water. To finish the chapter, our conclusions together with possible future work are given in Sect. 10.5.

10.2 Main Results

In this section we first discuss the canal dynamics assumed in the chapter. Then, we introduce our method for accelerating transport of water in a canal.

10.2.1 Preliminaries

Canal Dynamics

The canal dynamics are modelled by the so-called Saint Venant's equations [6, 14, 26]

$$\frac{\partial Q}{\partial x}+\frac{\partial A}{\partial t}=q_{\text{lat}},\qquad \frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q}{A}\right)^2+gA\frac{\partial h}{\partial x}+\frac{gQ|Q|}{C^2RA}=0, \tag{10.3}$$

where A is the cross-section area of the canal, q_{lat} is the lateral unitary net inflow, g is the gravitational acceleration, R is the hydraulic radius, and C is the Chézy constant. Water levels h and flows Q for the whole canal can be obtained from (10.3) when each pool is discretized longways into small section and (10.3) is integrated numerically for each section, assuming given flows through gates along the canal, yielding a precise model. That precise model is associated with high computational requirements and thus is not suited for use in a real-time controller design. Therefore, a simplified (linear) model of the canal can be obtained by a more coarse discretization of (10.3) into a water balance of each pool and a delay time in series [13, 20, 21, 27]:

$$h_i(k+1)=h_i(k)+\frac{T_{\text{m}}}{A_{\text{s},i}}(u_{i-1}(k-k_{\text{d}i})-u_i(k)+d_i(k)), \tag{10.4}$$

in which h_i is the water level at the downstream end of a pool, just before gate i, $k_{\text{d}i}$ is the time delay (in sampling steps) before an inflow from the upstream gate $i-1$ affects $h_i(k)$, T_{m} is the model sampling time (equal for all pools), $A_{\text{s},i}$ is the average surface area of pool i, d_i is the net inflow to pool i due to, e.g., an offtake ($d_i<0$) or rainfall ($d_i>0$), and u_i denotes the flow through gate i, with $u_0=Q_{\text{S}}$ denoting the inflow from the head gate. The flows $u_i(k)$, $i=1,\ldots,N$, are determined by the PI controllers installed at each gate according to the formula

$$u_i(k)=\max\left(u_i(k-1)+K_{\text{P},i}\left(e_i(k)-e_i(k-1)\right)+K_{\text{I},i}e_i(k),0\right), \tag{10.5}$$

in which $e_i=h_i(k)-h_i^{\text{ref}}(k)$ denotes the setpoint tracking error with $h_i^{\text{ref}}(k)$ being the value of a setpoint at sample step k, and $K_{\text{P},i}>0$ and $K_{\text{I},i}>0$ are the proportional and integral gains, respectively.

Illustrative Example: Characterization of CCID Main Canal

In this section we revisit the dynamical model of a canal given in Sect. 10.2.1 specifically for the representative example used later in the chapter to illustrate the functioning of the controller developed in the chapter. The representative example is the Central California Irrigation District Main Canal, which is a trapezoidal, gravity-flow channel, consisting of ten pools and ten control structures. To describe the canal in the case study, two models are used. The first one, which is the process model, is exactly the model (10.3) integrated using a longitudinal grid of 1 m, resulting in a very accurate model of the process. In that model, the standard PI controller as described earlier is enhanced to include a first-order low-pass filter as well. This means that to evaluate formula (10.5), we calculate the setpoint tracking errors $e_i(k)$ as $e_i(k)=h_i^{\text{filtered}}(k)-h_i^{\text{ref}}(k)$, where $h_i^{\text{filtered}}(k)=K_{\text{F},i}h_i^{\text{filtered}}(k-1)+(1-K_{\text{F},i})$

$h_i(k)$ with $K_{\mathrm{F},i} \in [0, 1)$. The flows through consecutive gates that are used as boundary conditions in model (10.3) obtained from formula (10.5) are further checked against physical restrictions of a gate (e.g., the maximum opening/width $\bar{\varphi}_i$ such that $0 \leq \varphi_i(k) \leq \bar{\varphi}_i$, and the maximum change rate $|\varphi_i(k) - \varphi_i(k-1)| \leq \Delta_{\max,\varphi,i}$), where $\varphi_i(k)$ denotes the position of gate i at step k. The resulting gate positions are given as

$$\varphi_i(k) = \begin{cases} \hat{\varphi}_i(k) & \text{if } \hat{\varphi}_i(k) \geq -\Delta_{\max,\varphi,i} + \varphi_i(k-1) \\ & \quad \text{and } \hat{\varphi}_i(k) \leq \Delta_{\max,\varphi,i} + \varphi_i(k-1), \\ \Delta_{\varphi,i} + \varphi_i(k-1) & \text{if } \hat{\varphi}_i(k) > \Delta_{\max,\varphi,i} + \varphi_i(k-1), \\ -\Delta_{\varphi,i} + \varphi_i(k-1) & \text{if } \hat{\varphi}_i(k) < -\Delta_{\max,\varphi,i} + \varphi_i(k-1), \end{cases} \tag{10.6}$$

in which $\hat{\varphi}_i(k) = \min\left(\max\left(\varphi_i(k), 0\right), \bar{\varphi}_i\right)$. Then, the flows u_i follow from [3]:

$$u_i(k) = c_i w_i \mu_i \varphi_i(k) \sqrt{2g(h_{i,\mathrm{up}}(k) - h_i^{\mathrm{crest}} - 1/2\varphi_i(k))}, \tag{10.7}$$

$$u_i(k) = c_i w_i \mu_i \sqrt{2g(h_{i,\mathrm{up}}(k) - h_{i,\mathrm{down}}(k))}, \tag{10.8}$$

$$u_i(k) = \frac{2}{3} c_i w_i \mu_i \sqrt{2/3g} \left(h_{i,\mathrm{up}}(k) - \varphi_i(k)\right)^{\frac{3}{2}}, \tag{10.9}$$

for a free-flowing undershot gate, a submerged undershot gate, and an overshot gate, respectively, where c_i denotes a calibration coefficient, w_i is the gate's width, μ_i is the contraction coefficient, and h_i^{crest} is the crest level (see [19] for extra details).

Next to the process model, which stems immediately from (10.3), we also consider a prediction model, which is a simplification of the process model. The simplification is done to relax the computational requirements that would otherwise be inordinate when the complex model (10.3) were used to derive the controller. As explained in Sect. 10.2.1, model (10.3) is linearized and the consequent model takes the form (10.4). This model is linear with the exception of the head gate settings $M_{\text{head gate}}$ that nonlinearly relate to the resulting flows Q_{S} [3].

We have now presented the prerequisites that form a foundation for the content of the further parts of the chapter. Subsequently, we give the main result communicated in the chapter: the derivation of the hierarchical controller used to accelerate transport of water in an irrigation channel.

10.2.2 Proposed Solution: Design of a Delivery Accelerating Hierarchical Controller

We now discuss the hierarchical controller proposed in the chapter. As mentioned earlier, the controller consists of two layers. The lower layer consists of local PI controllers, which operate continuously, and the higher layer includes a centralized

controller designed to work on an event-driven basis. This means that it is activated when there are events that require special actions to be taken on top of the control provided by the lower layer.

Denote by k_c the control step counter associated with the higher-layer controller, and similarly let T_c be the duration of the control cycle of the higher-layer controller, which is an integer multiple of the sampling time of the model T_m. In other words, $A_c = T_c/T_m \in \mathbb{N}$.

We split the derivation of the controller into two parts. First, we discuss the design assuming that only a single activation occurs and that the higher-layer controller may only be reactivated after the steady state is restored, see Sect. 10.2.2. Then, we elaborate on the necessary extensions to enable multiple concurrent active periods of the higher layer, see Sect. 10.2.2.

Concept Description: Single Activation

In this section we introduce the basic ideas regarding the hierarchical controller, where we assume, for the sake of simplifying the message, that only a single activation of the higher layer of the hierarchical controller takes place. At the beginning, before the hierarchical controller is ready to work, a design step needs to be performed, in which initial proportional and integral parameters of the PI controllers are selected. Various methods can be used for this, e.g., a manual tuning, Ziegler-Nichols tuning method [29], lambda tuning method [7], or optimization-based tuning. Here, we briefly explore the last option, i.e., we propose to find the gains through the minimization of a cost function accounting for deviations in the water levels with respect to their setpoints and immoderate fluctuations in water levels and flows, assuming an input consisting of a selection of representative subscenarios with a certain number of flow changes in the pools and changes to the head gate:

$$J_{\text{pre}} = \sum_{i=1}^{N} \sum_{j=1}^{N_{\text{tot}}} \left(w_e e_i(j) + w_{\Delta h}(h_i(j) - h_i(j-1)) + w_{\Delta u}(u_i(j-1) - u_i(j-2)) \right), \tag{10.10}$$

in which $u_i(-1)$ is assumed as a given initial condition describing past flows, and N_{tot} is the length of the prediction horizon in the preliminary step, which can differ from the length of the prediction horizon N_p used by the higher-layer controller in its regular operation mode. In the preliminary step, the cost function J_{pre} is minimized with respect to the proportional and integral gains of the PI controllers, which means that the preliminary optimization problem can be stated as

$$(K_{P,i}^*, K_{I,i}^*) = \min_{K_{P,i}, K_{I,i}} J_{\text{pre}}, \quad \text{subject to } K_{v,i}^{\min} \le K_{v,i} \le K_{v,i}^{\max}, \text{ (10.4)}, \text{ (10.5)}, \tag{10.11}$$

where $\nu \in \{\mathrm{P}, \mathrm{I}\}$, $K_{\nu,i}^{\min} > 0$ are small constants ensuring all gains are strictly positive, and $K_{\nu,i}^{\max} > K_{\nu,i}^{\min}$ is an upper bound for the gains. After solving (10.11), values $K_{\mathrm{P},i}^{*}$ and $K_{\mathrm{I},i}^{*}$ are used later on by the PI controllers.

Now, suppose that the activation of the higher-layer controller takes place at sampling step $k_{\text{activation}}$. Then, the movement of the head gate for the next $N_{\mathrm{p}} A_{\mathrm{c}}$ steps can be found using (10.1) according to the relation

$$M_{\text{head gate}}(k_{\text{activation}} + jA_{\mathrm{c}} + \ell) = M_{\text{head gate}}^{\text{control}}(j), \text{ for } \begin{array}{l} \ell = 0, \ldots, A_{\mathrm{c}} - 1, \\ j = 0, \ldots, N_{\mathrm{p}} - 1 \end{array} \tag{10.12}$$

for $k = k_{\text{activation}}, k_{\text{activation}} + 1, \ldots, k_{\text{activation}} + N_{\mathrm{p}} A_{\mathrm{c}} - 1$ and

$$M_{\text{head gate}}(k) = M_{\text{head gate}}^{\text{steady state}} = M_{\text{head gate}}^{\text{control}}(N_{\mathrm{p}} - 1),$$

for $k > k_{\text{activation}} + N_{\mathrm{p}} A_{\mathrm{c}} - 1$. As can be seen above, after $N_{\mathrm{p}} A_{\mathrm{c}}$ steps, the head gate settings are set to a new steady-state level

$$M_{\text{head gate}}^{\text{steady state}} = M_{\text{head gate}}^{\text{control}}(N_{\mathrm{p}} - 1)$$

afterwards.

In contrast to the head gate settings, which may end up with a different steady-state value after the activation, the setpoints of the local controllers are only changed temporarily to enable speedy flow changes and afterwards return to their predefined levels. Specifically, after the activation, the setpoints of the local controllers change using the triple $(H^{\text{ref, delivery}}, T^{\text{on}}, T^{\text{off}})$ (10.2). For each pool $i = 1, \ldots, N$, the setpoints are found as

$$h_i^{\text{ref}}(k) = \begin{cases} h_i^{\text{ref, delivery}} & \text{if } k_i^{\text{on}} \le k \le k_i^{\text{off}}, \\ h_i^{\text{ref, normal}} & \text{otherwise,} \end{cases} \tag{10.13}$$

in which $k_i^{\text{on}} = \left\lceil \frac{t_i^{\text{on}}}{T_{\mathrm{m}}} \right\rceil$ and $k_i^{\text{off}} = \left\lceil \frac{t_i^{\text{off}}}{T_{\mathrm{m}}} \right\rceil$. Expression (10.13) ensures that stepwise setpoint modifications are executed.

We now discuss what is different in cases A) and B).

A) In this case, the hierarchical controller is directly ready for normal operation. For each activation of the higher-layer controller during its regular operation, the control input that needs to be found is the quadruple

$$\mathcal{U}_{\mathrm{A}} = \left(\tilde{M}_{\text{head gate}}^{\text{control}}, H^{\text{ref, delivery}}, T^{\text{on}}, T^{\text{off}} \right). \tag{10.14}$$

B) Now, in addition to modifying the head gate settings and the setpoints, also the gains of the PI controllers are temporarily changed using a quadruple

$(K_{\mathrm{P}}^{\mathrm{temp}}, K_{\mathrm{I}}^{\mathrm{temp}}, T^{\mathrm{on,gains}}, T^{\mathrm{off,gains}})$. The gains are then changed in the system in a stepwise manner, very much similar to how the setpoints are changed:

$$K_{\nu,i}(k) = \begin{cases} K_{\nu,i}^{\mathrm{temp}} & \text{if } k_i^{\mathrm{on,gains}} \leq k \leq k_i^{\mathrm{off,gains}}, \\ K_{\nu,i}^{\mathrm{normal}} & \text{otherwise,} \end{cases} \tag{10.15}$$

where $\nu \in \{\mathrm{P}, \mathrm{I}\}$ and $k_i^{\mathrm{on,gains}} = \left\lceil \frac{t_i^{\mathrm{on,gains}}}{T_{\mathrm{m}}} \right\rceil$ and $k_i^{\mathrm{off,gains}} = \left\lceil \frac{t_i^{\mathrm{off,gains}}}{T_{\mathrm{m}}} \right\rceil$. The overall control input to be found in case B) is denoted with $\mathcal{U}_{\mathrm{B}}$, which is a tuple defined as

$$\mathcal{U}_{\mathrm{B}} = \left(\tilde{M}_{\mathrm{head\,gate}}^{\mathrm{control}}, H^{\mathrm{ref,\,delivery}}, T^{\mathrm{on}}, T^{\mathrm{off}}, K_{\mathrm{P}}^{\mathrm{temp}}, K_{\mathrm{I}}^{\mathrm{temp}}, T^{\mathrm{on,gains}}, T^{\mathrm{off,gains}} \right). \tag{10.16}$$

With the control inputs $\mathcal{U}_{\mathrm{A}}$ and $\mathcal{U}_{\mathrm{B}}$ defined for cases A) and B), respectively, we may now move forward to defining the cost function that is to be minimized. According to the objectives mentioned in short in Sect. 10.1.2, we define the cost function as follows

$$\begin{aligned} J = {} & \alpha \sum_{j=1}^{A_{\mathrm{c}} N_{\mathrm{p}}} \left(u_N \left(k_{\mathrm{activation}} + j - 1 \right) - Q_{\mathrm{S,\,base}} \right)^2 \\ & + \beta \sum_{i=1}^{N} \sum_{j=1}^{A_{\mathrm{c}} N_{\mathrm{p}}} \left(h_i \left(k_{\mathrm{activation}} + j \right) - h_i^{\mathrm{ref}} \left(k_{\mathrm{activation}} + j \right) \right)^2 \\ & + \mu \sum_{i=1}^{N} \left(t_i^{\mathrm{off}} - t_i^{\mathrm{on}} \right)^2 + \zeta \sum_{i=1}^{N} \left(\Delta h_i^{\mathrm{ref}} \right)^2 + \lambda_\xi \sum_{i=1}^{N} \left(t_i^{\mathrm{off,gain}} - t_i^{\mathrm{on,gain}} \right)^2, \end{aligned} \tag{10.17}$$

where α, β, γ_1, γ_2, μ and ζ are positive weighting coefficients, $\xi \in \{\mathrm{A}, \mathrm{B}\}$, $\lambda_{\mathrm{A}} = 0$, $\lambda_{\mathrm{B}} > 0$, and $\Delta h_i^{\mathrm{ref}} = h_i^{\mathrm{ref,\,delivery}} - h_i^{\mathrm{ref,\,normal}}$. The form of the cost function is chosen in such a specific way to capture the objectives of the controller. In particular, the smaller the first term of J in (10.17) is, the less disruption is for the further downstream users beyond the stretch of the canal under consideration with N pools, as the inflow to that further part is closer to the ordained base flow $Q_{\mathrm{S,\,base}}$. Furthermore, the second term (10.17) penalizes the setpoint tracking errors in order to refrain from selecting control actions resulting in undue deviations in the water levels with respect to their respective setpoints. Thirdly, terms in (10.17) facilitate switching the setpoints and the PI gains in case B) promptly back to their normal levels to bring the system to the normal operating conditions after an activation apace.

To ensure adequate operation of the controller, the following hard constraints need to be satisfied for all $i \in \{1, \ldots, N\}$:

$$h_i^{\min} \leq h_i(k_{\text{activation}} + j) \leq h_i^{\max}, \qquad j = 1, \ldots, N_\text{p} A_\text{c}, \tag{10.18}$$

$$h_i^{\min} \leq h_i^{\text{ref}}(k_{\text{activation}} + j) \leq h_i^{\max}, \qquad j = 1, \ldots, N_\text{p} A_\text{c}, \tag{10.19}$$

$$t_i^{\text{off}} \geq t_i^{\text{on}} + T_\text{m}, \tag{10.20}$$

$$t_i^{\text{on}} \geq k_{\text{activation}} T_\text{m}, \tag{10.21}$$

$$t_i^{\text{off,gain}} \geq t_i^{\text{on,gain}} + T_\text{m}, \tag{10.22}$$

$$t_i^{\text{on,gain}} \geq k_{\text{activation}} T_\text{m}, \tag{10.23}$$

$$M_{\text{head gate}}^{\min} \leq M_{\text{head gate}}(k_{\text{activation}} + j) \leq M_{\text{head gate}}^{\max}, \; j = 0, \ldots, N_\text{p} A_\text{c} - 1, \tag{10.24}$$

$$|M_{\text{head gate}}(k_{\text{activation}} + j) - M_{\text{head gate}}(k_{\text{activation}} + j - 1)| \leq \Delta M_{\text{head gate}}^{\max}, \tag{10.25}$$

$$j = 0, \ldots, N_\text{p} A_\text{c} - 1,$$

$$K_{\nu,i}^{\min} \leq K_{\nu,i} \leq K_{\nu,i}^{\max}, \tag{10.26}$$

$$t_i^{\text{off}} \leq k_{\text{activation}} T_\text{m} + T_\text{c} N_\text{c}, \tag{10.27}$$

$$t_i^{\text{off,gains}} \leq k_{\text{activation}} T_\text{m} + T_\text{c} N_\text{c}, \tag{10.28}$$

$$M_{\text{head gate}}^{\text{control}}(k_{\text{activation}} + j) = M_{\text{head gate}}^{\text{control}}(N_\text{c} - 1), \qquad j = N_\text{c}, \ldots, N_\text{p} - 1, \tag{10.29}$$

$$M_{\text{head gate}}(k_{\text{activation}} + jA_\text{c}) = M_{\text{head gate}}^{\text{control}}(N_\text{p} - 1), \; j > N_\text{p} - 1, \tag{10.30}$$

in which $N_\text{c} \leq N_\text{p}$ is the control horizon. Constraints (10.18) and (10.19) correspond to the physical constraints of the depth of the canal. Constraints (10.20) and (10.21) ensure that the first switch of the setpoint t_i^{on} occurs after the activation moment of the higher-layer controller and the second moment t_i^{off} occurs at least one sampling step after the first one. Constraints (10.22) and (10.23) work similarly but for the PI gains. Clearly, they only apply in case B). Furthermore, constraints (10.24) and (10.25) make sure that the movements of the head gate are compliant with its minimum and maximum position and its maximum change rate, respectively, and constraint (10.26) deals with the upper and lower bounds for the proportional and integral gains. This latter constraint yet again is valid in case B) only. Next, constraints (10.27)–(10.29) enforce all changes to be executed within the control horizon, and, lastly, constraint (10.30) introduces new steady-state head gate settings after the activation to be equal to the last component of $\tilde{M}_{\text{head gate}}^{\text{control}}$, i.e., $M_{\text{head gate}}^{\text{control}}(N_\text{p} - 1)$.

Extension: Multiple Activations

So far, we have discussed the design of the hierarchical controller for a single active period at a time, and a re-activation only allowed after the steady-state has been restored after a preceding activation. Here, we give a broad overview of how the event-driven controller design needs to be extended to enable multiple concurrent active periods.

First, we discuss how the higher-layer controller is allowed to be activated. We introduce two options: the synchronous one and the asynchronous one. In the synchronous case, see Fig. 10.1, the activation can only occur at a multiple of the control step k_c. If there are events in between two control steps k_c and $k_c + 1$, they are grouped together and sent jointly to the higher-layer controller. In contrast, in the asynchronous case, see Fig. 10.2, the activation can occur at any sample step k. Here, we propose to introduce a short time window δ in which individual events occurring within δ time units after a first one in a round are grouped and conveyed to the higher-layer controller together. δ is a design parameter to be selected by the operator of the canal enabling events happening soon after each other being rendered as a single event and swiftly sent to the higher layer controller. We also impose a minimal reactivation time of T_c time units so that the activations do not occur too frequently.

Now, we analyze how the setpoints are permitted to change. Denote by $s \in \mathbb{Z}$ the activation counter, which is initiated with 0 and is incremented every time a new activation takes place. Two options to change the setpoints are the block-modifying strategy and the block-adding strategy. The block-modifying strategy is illustrated in Fig. 10.3 and rely on the principle that once a setpoint block is started

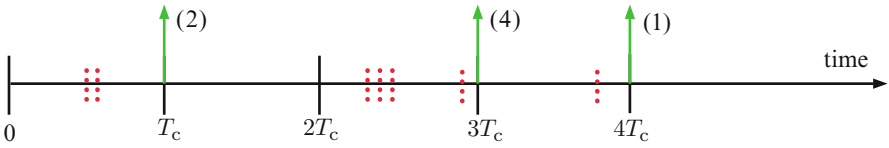

Fig. 10.1 Activation of the higher-layer controller in the synchronous case. All individual events are denoted with *dotted bars*. *Arrows* are used to indicate when the activation occurs with a label representing how many events are dealt with during each activation

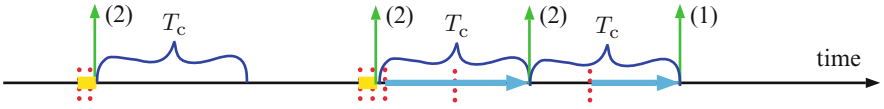

Fig. 10.2 Activation of the higher-layer controller in the asynchronous case. All individual events are denoted with *dotted bars*. *Vertical arrows* are used to indicate when an activation occurs with a label representing how many events are dealt with during each activation. *Horizontal arrows* show the delays with activation of the higher-layer controller for individual requests because of the minimum interval between activations of T_c. *Horizontal bars* indicate the length of the time window δ used to accumulate events occurring soon after each other

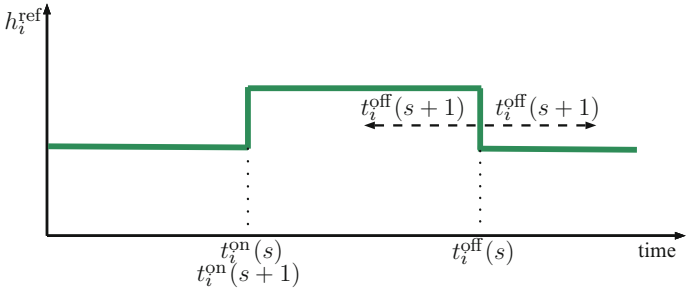

Fig. 10.3 Possible setpoint profiles using block-modifying formulation

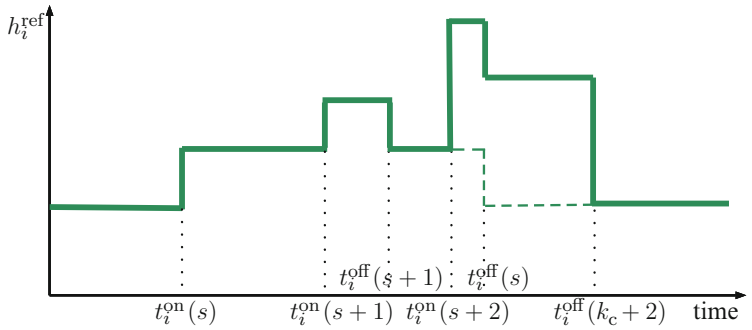

Fig. 10.4 Possible setpoint profiles using block-adding formulation

for activation s, it may only be changed in activation $s+1$ by changing its length, i.e., by modifying $t_i^{\mathrm{off}}(s) = t_i^{\mathrm{off}}(s+1)$. So the overall setpoint profiles, when the block-modifying strategy is implemented, consist of blocks next to one another: when one block finishes, a new one can start. Conversely, in the block-adding strategy, in every new activation new blocks are added on top of the preexisting ones. This is depicted in Fig. 10.4.

In a similar way also the proportional and integral gains of the local controllers may be changed in case B) using block-adding or block-modifying formulations. We leave out the details here for the sake of page limitations.

Next, we discuss how the head gate is controlled by the higher-layer controller for multiple concurrent active periods of the higher-layer controller. A major factor that determines the functioning of the head gate is that when the s^{th} activation of the higher-layer controller takes place at sample step $k_{\mathrm{activation},s}$, the profile of the head gate found for the previous activation is to be updated and a new profile is found and executed. In doing so, the higher-layer controller is of course also aware of the events occurring before the present activation s and the new profile also accounts for them. We illustrate this in Fig. 10.5, where four activations are shown and depicted with the arrows. Observe that once an activation s takes place, the previously found head gate profile $\tilde{M}_{\mathrm{head\,gate}}^{\mathrm{control}}(s-1)$ is no longer valid and the new profile $\tilde{M}_{\mathrm{head\,gate}}^{\mathrm{control}}(s)$

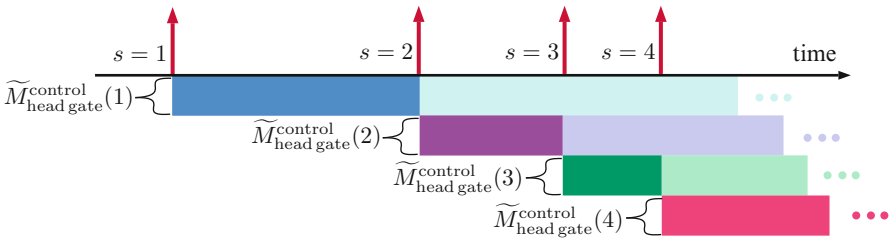

Fig. 10.5 Control of the head gate settings as administrated by the higher-layer controller in face of multiple concurrent activations. Four activations are shown and indicated with the *arrows* and the corresponding profiles for the head gate settings $\tilde{M}^{\text{control}}_{\text{head gate}}(s)$

is applied. This profile is executed until another activation $s + 1$ of the higher-layer controller occurs, at which point the profile of the head gate settings $\tilde{M}^{\text{control}}_{\text{head gate}}(s)$ is again updated with the newly found profile $\tilde{M}^{\text{control}}_{\text{head gate}}(s + 1)$.

The above discussion presents the implementation principles of the hierarchical controller presented in the chapter. The main alterations that need to be accounted for in the multiple-activation case with respect to the single activation is how to deal with modifying setpoints and changing PI gains in case B). This needs to be decided in a design phase, based on the performance requirements of a specific application. The controller then works generally as follows: when there is a need for an activation, the higher-layer controller is triggered and the changes are applied to the system. Then, whenever T_c time units of the reactivation time has passed and new events occur, the controller can be re-activated, and so on.

10.3 Illustrative Example: Results

In this section we discuss the application of the proposed controller on an accurate numerical model of Central California Irrigation District Main Canal. We validate the prediction model used in the controller design in Sect. 10.3.1. Then in Sect. 10.3.2 we present the results obtained by applying the controller to CCID Main Canal and analyze the performance obtained.

10.3.1 Validation of the Prediction Model

We validate pool by pool the prediction model (10.4) with for the parameters given in Table 10.1, where $k_{\text{d}i}$ is given in sampling steps, $A_{\text{s},i}$ in m^2, and T_{m} is 1 min. For each pool $i = 1, \ldots, N$ individually, we apply a step increase of $1.5\,\text{m}^3/\text{s}$ in the

Table 10.1 Parameters of the prediction model (10.4)

i	1	2	3	4	5	6	7	8	9	10
$k_{\mathrm{d}i}$	13	26	21	25	13	11	44	57	15	36
$A_{\mathrm{s},i}$	154842	132722	121181	154842	73346	58066	163951	139358	67979	154842

Table 10.2 VAF for the responses of the process model and the prediction model

i	1	2	3	4	5	6	7	8	9	10
VAF	94%	88%	91%	88%	97%	97%	93%	87%	96%	86%

upstream inflow u_{i-1} with the outflow u_i from the pool at the downstream end kept constant, and with a base flow of $Q_{\mathrm{S,base}} = 7.14\,\mathrm{m}^3/\mathrm{s}$. We apply the same stimuli both for the prediction model and the precise numerical model of the canal. The resulting water levels h_i are measured and compared using the variance-accounted-for criterion:

$$\mathrm{VAF}(m,n) = \left(1 - \frac{\mathrm{var}(m-n)}{\mathrm{var}(m)}\right) \cdot 100\,\%, \tag{10.31}$$

where signal n is obtained from the accurate simulator and signal m from the prediction model. We give the corresponding VAF values for each pool in Table 10.2. It is concluded that the prediction model closely matches the accurate process model and so it is deemed to be suitable for the purpose of the derivation of the controller.

10.3.2 Control Results

We now illustrate the functioning of the proposed control approach in a simulation-based case study. We study here a representative example of a single flow change in pool 10 of magnitude $2.5\,\mathrm{m}^3/\mathrm{s}$ with a base flow of $Q_{\mathrm{S,base}} = 7.14\,\mathrm{m}^3/\mathrm{s}$. The flow change occurs at step $k_{\mathrm{known}} = 180$ and is not announced beforehand. The two cases A) and B) are shown and compared with each other and with a standard method[2] currently used in the field. Using the standard method in the aforementioned circumstances introduces a big disturbance in outflow from pool 10 as in face of the increased offtake outflow from pool 10, the water levels in pool 10 start to decrease and so the PI controller in that pool decreases the outflow to maintain the water level at its desired setpoint. As explained in Sect. 10.2.2, this is an undesirable situation, which is accounted for in the objective function (10.17) of

[2]The standard method works by increasing inflow from the head gate at the time when an offtake is announced to provide extra water needed for the additional offtake and letting the PI controllers transport that water through subsequent pools to the offtake point. This method relies on the PI controllers only with the higher-layer controller being absent.

the hierarchical controller. Therefore, we expect that the situation can be improved when employing the hierarchical controller introduced in this chapter.

To compare the performance when different control methods are applied, we introduce an a posteriori cost function defined as

$$J_{\text{post}} = \alpha \sum_{j=k_{\text{known}}}^{N_{\text{F}}} \left(u_N\left(j-1\right) - Q_{\text{S, base}}\right)^2 + \beta \sum_{i=1}^{N} \sum_{j=k_{\text{known}}}^{N_{\text{F}}} e_i^2\left(j\right), \tag{10.32}$$

where $N_{\text{F}} = 780$ marks the duration of the simulation and the weighting parameters are the same as in the cost function J (10.17) used to derive the controller. These parameters are $\alpha = 5$, $\beta = 10$, $\gamma = 1$, $\mu = 1$, $\zeta = 1$, and $\lambda_B = 1$. Moreover, we use $T_{\text{m}} = 1\,\text{min}$, $A_{\text{c}} = 15$, $N_{\text{c}} = 16$ (corresponding to 240 sample steps), and $N_{\text{p}} = 36$ (i.e., 540 sample steps). In addition, the upper bounds for the proportional gains in the design stage of the hierarchical controller as well as in the normal operation in case B) are determined through stability analysis and subsequently $K_{\text{P},i}^{\max}$ for $i = 1, \ldots, N$ is 89.17, 314.16, 285.47, 363.50, 177.71, 141.57, 416.09, 336.24, 159.90, and 363.50 respectively. The upper bounds for the integral gains are selected as $K_{\text{I},i}^{\max} = K_{\text{P},i}^{\max}$ and the lower bounds for both the proportional and integral gains are $K_{\text{P},i}^{\min} = K_{\text{I},i}^{\min} = 0.01$.

In the design stage, the weighting parameters are chosen to be $w_{\text{e}} = 0.001$, $w_{\Delta h} = 250$ and $w_{\Delta u} = 500$, and the prediction horizon is chosen to be equal to the prediction horizon in the normal operation of the higher-layer controller, i.e., $N_{\text{tot}} = N_{\text{p}} = 540$ sample steps. The scenario considered to determine the gains in the design step consists of five changes in offtakes along the canal and five changes to the head gate to compensate for the offtakes. The resulting gains $K_{\text{P},i}^{*}$ and $K_{\text{I},i}^{*}$ used later on throughout the simulation are given in Table 10.3. Note that in the standard method different gains are used (see $K_{\text{P},i}^{\text{standard}}$ and $K_{\text{I},i}^{\text{standard}}$ in Table 10.3) and these are the gains currently implemented in the real CCID Main Canal.

The results obtained are given in Table 10.4, which lists the values of the a posteriori cost function J_{post} for the three control approaches considered: the standard method, the new hierarchical controller for case A), and the new hierarchical controller for case B). It can be immediately seen that the standard method performs inferiorly in comparison to the new hierarchical controller. The corresponding value

Table 10.3 Proportional and integral gains

	1	2	3	4	5	6	7	8	9	10
$K_{\text{P},i}^{*}$	47.29	207.05	273.25	262.83	71.31	117.79	56.12	20.30	13.68	59.32
$K_{\text{I},i}^{*}$	8.30	2.42	2.98	4.06	4.41	20.26	5.68	0.41	0.28	14.39
$K_{\text{P},i}^{\text{temp}}$	43.58	156.40	80.15	132.43	38.60	25.77	70.52	31.16	19.20	35.59
$K_{\text{I},i}^{\text{temp}}$	9.36	1.38	3.64	3.56	1.39	0.69	2.74	0.47	0.24	3.18
$K_{\text{P},i}^{\text{standard}}$	186	157	143	182	190	152	208	168	165	182
$K_{\text{I},i}^{\text{standard}}$	0.5	0.6	0.6	0.7	1.0	1.1	0.9	1.0	1.8	1.8

Table 10.4 Comparison between the values of J_{post} obtained with the standard method and with the hierarchical controller

	Standard method	Proposed method—case A)	Proposed method—case B)
J_{post}	10800.1	3457.9	1964.5

of the a posteriori cost function is $J_{\text{post}} = 10{,}800.1$. When the new hierarchical controller is applied for case A), a better performance is achieved with the value of the a posteriori cost function being over three times smaller, i.e., $J_{\text{post}} = 3{,}457.9$. This is because the hierarchical controller explicitly takes into account the objectives of the control design and works in a way to minimize the deviation of the system behavior against the desired behavior described using the cost function. The best performance, though, is obtained with the new hierarchical controller for case B), with the a posteriori cost function $J_{\text{post}} = 1{,}964.5$ equating to less than a fifth of the cost function achieved with the standard method. The difference in performance of the controller in cases A) and B) is due to the fact that in case A) the gains of the local PI controllers are fixed and are not selected nor updated for the specific behavior of the system with setpoint changes etc. Conversely, in case B) the controller has the extra freedom to modify the proportional and integral gains on the spot for the purpose of meeting the objectives defined in the cost function. Consequently, in the transient time of dealing with the sudden offtake in pool 10, the PI gains are changed (mostly lowered) to ensure a gentler reaction of the PI controllers to the changed setpoints and thus smoother flows through gates are obtained.

10.4 Linking Transport *of* Water with Transport *over* Water

In the work presented here we explicitly work towards methods for transport *of* water to be delivered to farmers through an irrigation canal. However, the proposed solution also implicitly enhances transport *over* water through maintaining water levels in waterways within a certain range so that ships can operate safely. As such the control strategy proposed in Sect. 10.2.2 can also ensure transport over water. To that end, a number of aspects have to taken care of. In particular, the constraints on the maximum and minimum allowable water levels in (10.18) and (10.19) need to be assigned accordingly.

In future work, the link between the transport-of-water component, as examined in the chapter, and transport over water could be further explored and studied more explicitly. One possible approach to introduce an explicit treatment of transport over water within the presented framework could be to explore a multi-objective approach including the objective of transport of water and the objective of transport over water. The unified control scheme could be formulated in a centralized or distributed fashion (cf. [23] where different objectives are associated with different agents) so as to enable both transport *of* water and transport *over* water.

10.5 Conclusions and Future Research

This chapter has presented a hierarchical controller for the purpose of accelerating transport of water in an irrigation canal. The transport is accelerated in the sense that flow changes along the canal can occur more rapidly, thus more efficient control performance is obtained. The hierarchical controller consists of two layers. In the lower layer, local decentralized PI controllers take care of the canal in its normal operation. In contrast, the higher layer is invoked in response to events only and comprises a centralized predictive controller. This controller works by providing the head gate with an updated profile to account for the event as well as by transiently altering the settings of the local PI controllers. This introduces extra buffer where water can be temporarily stored or can be borrowed from for a speedy delivery to the location where water is needed. We have given in the chapter an in-depth description of the controller design, which includes basic operational concepts and the necessary extensions for continuous operation with multiple events activating the higher-layer controller. The performance of the hierarchical control approach has been demonstrated in a simulation-based case study, which shows that the hierarchical controller outperforms the standard method.

In addition to extending the current work by combining explicitly transport of water with transport over water, further open topics include robustness analysis and robust design in face of uncertainties or unmeasured disturbances. Moreover, an analysis of computational requirements of the proposed control approach to make sure that the scheme can be employed in a real-time operation remains to be studied.

Acknowledgements Research supported by the European Union Seventh Framework Programme [FP7/2007–2013] under grant agreement no. 257462 HYCON2 Network of Excellence.

References

1. Álvarez A, Ridao M, Ramirez D, Sánchez L. Constrained predictive control of an irrigation canal. J Irrig Drain Eng. 2013;139(10):841–854.
2. Bemporad A, Morari M. Control of systems integrating logic, dynamics, and constraints. Automatica 1999;35(3):407–427.
3. Bos G. Discharge measurement structures. In: International Institute for Land Reclamation and Improvement, Publication 20. ILRI; 1976.
4. Camacho EF, Bordons C. Model predictive control. Berlin Heidelberg: Springer; 1999.
5. Cantoni M, Weyer E, Li Y, Ooi SK, Mareels I, Ryan M. Control of large-scale irrigation networks. In: Proc IEEE. 2007;95(1):75–91.
6. Chow VT (1959) Open-channel hydraulics. McGraw-Hill civil engineering. London: McGraw-Hill; 1959.
7. Dahlin EB. Designing and tuning digital controllers. Instrum Control Syst. 1968;41(6):77–83.
8. De Schutter B, De Moor B. Optimal traffic light control for a single intersection. Eur J Control. 1998;4(3):260–276.
9. Li Y, Cantoni M (2008) Distributed controller design for open water channels. In: Proceedings of the 17th IFAC World Congress, Seoul; 2008. pp. 10033–10038

10. Litrico X, Fromion V, Baume J-P, Rijo M. Modelling and PI controller design for an irrigation canal. In: Proceedings of the 2003 European Control Conference, Cambridge; 2003.
11. Litrico X, Malaterre P-O, Baume J-P, Vion P-Y, Ribot-Bruno J. Automatic tuning of PI controllers for an irrigation canal pool. J Irrig Drain Eng ASCE. 2007;133:27–37.
12. Maciejowski JM. Predictive control with constraints. Essex: Prentice Hall; 2002.
13. Malaterre P-O. Control of irrigation canals: why and how? In: Proceedings of the international workshop on numerical modelling of hydrodynamics for water resources, Zaragoza; 2007. pp. 271–293.
14. Malaterre P-O, Baume JP. Modeling and regulation of irrigation canals: existing applications and ongoing researches. In: Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, San Diego; 1998. pp 3850–3855.
15. Negenborn RR, van Overloop P-J, Keviczky T, De Schutter B. Distributed model predictive control for irrigation canals. Netw Heterog Media. 2009;4(2):359–380.
16. Ooi SK, Weyer E. Control design for an irrigation channel from physical data. Control Eng Pract. 2008;16(9):1132–1150.
17. Perkins S. Is agriculture sucking fresh water dry? Science NOW, online. Published 13 Feb 2012.
18. Sadowska A, De Schutter B, van Overloop P-J. Delivery-oriented hierarchical predictive control of an irrigation canal: event-driven versus time-driven approaches. IEEE Trans Control Syst Technol. 2015, doi: 10.1109/TCST.2014.2381600
19. Sadowska A, van Overloop P-J, Burt C, De Schutter B. Hierarchical operation of water level controllers: formal analysis and application on a large scale irrigation canal. Water Resour Manag. 2014;28(14):4999–5019
20. Schuurmans J, Clemmens A, Dijkstra S, Hof A, Brouwer R. Modeling of irrigation and drainage canals for controller design. J Irrig Drain Eng. 1999;125(6):338–344.
21. Schuurmans J, Hof A, Dijkstra S, Bosgra O, Brouwer R. Simple water level controller for irrigation and drainage canals. J Irrig Drain Eng. 1999;125(4):189–195.
22. Silva P, Botto MA, Figueiredo J, Rijo M. Model predictive control of an experimental water canal. In: Proceedings of the 2007 European Control Conference, Kos; 2007. pp. 2977–2984.
23. Tian X, Maestre JM, van Overloop PJ, Negenborn RR. Distributed model predictive control for multi-objective water system management. In: Proceedings of the 10th International Conference on Hydroinformatics, Hamburg, July 2012. Paper 175.
24. van Ekeren H, Negenborn RR, van Overloop PJ, De Schutter B. Hybrid model predictive control using time-instant optimization for the Rhine-Meuse delta. In: Proceedings of the 2011 IEEE International Conference on Networking, Sensing and Control, Barcelona; 2011. pp. 216–221.
25. van Ekeren H, Negenborn RR, van Overloop PJ, De Schutter B. Time-instant optimization for hybrid model predictive control of the Rhine-Meuse Delta. J. Hydroinformatics. 2013;15(2):271–292.
26. van Overloop PJ, Clemmens AJ, Strand RJ, Wagemaker RMJ. Real-time implementation of model predictive control on MSIDD's WM canal. J Irrig Drain Eng ASCE. 2010;136(11): 747–756.
27. van Overloop PJ, Schuurmans J, Brouwer R, Burt C. Multiple-model optimization of proportional integral controllers on canals. J Irrig Drain Eng ASCE. 2005;131(2):190–196.
28. Xu M, Negenborn RR, van Overloop PJ, van de Giesen NC. De Saint-Venant equations-based model predictive control of open channel flow. Adv Water Res. 2012;49:37–45.
29. Ziegler JG, Nichols NB. Optimum Settings for Automatic Controllers. Trans ASME. 1942;64:759–768.